

## PRINT SYSTEM

(2)

Patent number: JP10124270  
 Publication date: 1998-05-15  
 Inventor: HONMA HIDEO  
 Applicant: CANON INC  
 Classification:  
 - international: G06F3/12; B41J29/38  
 - european:  
 Application number: JP19960280459 19961023  
 Priority number(s):

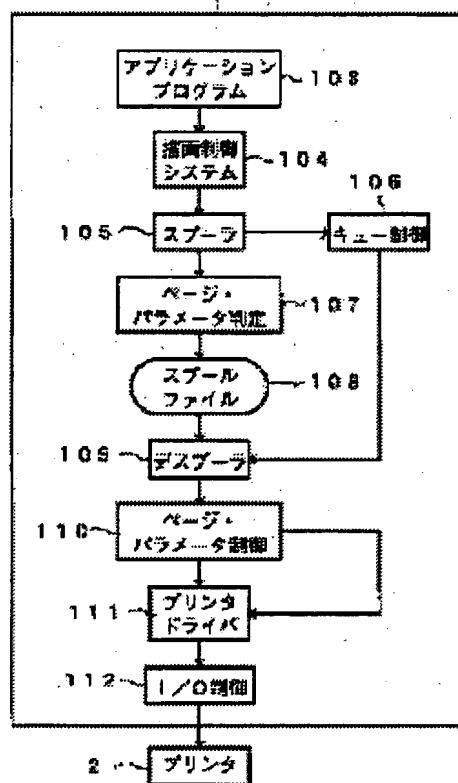
Also published as:

 JP10124270 (A)

## Abstract of JP10124270

**PROBLEM TO BE SOLVED:** To obtain a print system capable of automatically changing printer processing in accordance with an attribute of data to be plotted by every page.

**SOLUTION:** An application 103 successively transfers generated print job data to a plotting control system 104. The plotting control system 104 transfers the print job data to a spooler 105 by converting the print job data into a form in which a plotting processing is performed. The spooler 105 has a queue control 106 perform queuing processing of a print job. A page parameter judging part 107 implants a parameter to instruct plotting by every page into a spool file. A despooler 109 takes the spool file out of the print job and transfers the file to a page parameter control part 110. The page parameter control part 110 checks contents of the print job and instructs plotting in accordance with a plotting instruction by every page of the job.



Data supplied from the esp@cenet database - Patent Abstracts of Japan

特開平10-124270

(43)公開日 平成10年(1998)5月15日

(51) Int.Cl.<sup>6</sup>  
G 0 6 F 3/12  
B 4 1 J 29/38

識別記号

F 1  
G 0 6 F 3/12  
B 4 1 J 29/38B  
Z

審査請求 未請求 請求項の数12 O.L (全 13 頁)

(21)出願番号 特願平8-280459

(22)出願日 平成8年(1996)10月23日

(71)出願人 00000107

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 本間 英雄

東京都大田区下丸子3丁目30番2号 キヤ  
ノン株式会社内

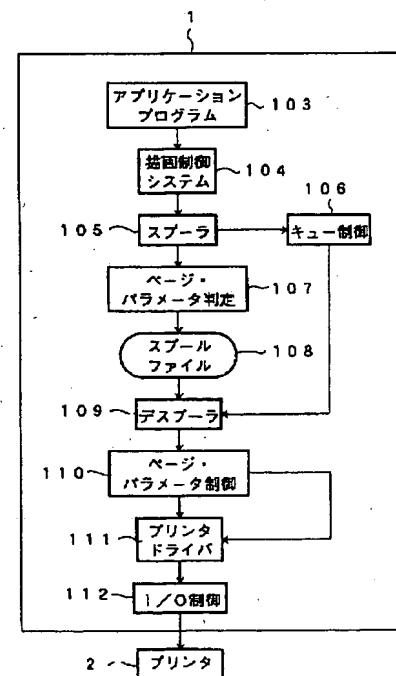
(74)代理人 弁理士 谷 義一 (外1名)

(54)【発明の名称】 プリント・システム

## (57)【要約】

【課題】 ページごとの描画しようとするデータ属性に応じてプリント処理を自動的に変えることができるプリント・システム

【解決手段】 アプリケーション103は、生成したプリント・ジョブ・データを描画制御システム104に順次渡す。描画制御システム104は、プリント・ジョブ・データを描画処理できる形式に変換してスプーラ105に渡す。スプーラ105は、キュー制御106でプリント・ジョブのキューイング処理を行わせる。ページ・パラメータ判定部107は、ページ毎の描画指示を行うパラメータをスプール・ファイル108の中に埋め込む。デスプーラ109はプリント・ジョブのスプール・ファイルを取りだし、ページ・パラメータ制御部110へ渡す。ページ・パラメータ制御部110は、プリント・ジョブの内容をチェックし、そのジョブの各ページ毎に描画指示に従って描画の指示を行う。



## 【特許請求の範囲】

【請求項1】 ホスト・コンピュータ上における、アプリケーションの生成した描画データからプリンタへのプリント・データを形成し、プリンタに転送して印刷するプリント処理を含むプリント・システムにおいて、前記プリント処理は、

前記描画データの各ページ内容を検出し、検出した結果に応じて、各ページごとの描画モードを判定し、

各ページごとに描画モードを指定するパラメータを生成し、

生成した前記パラメータを保持し、各ページ処理に同期して、保持していた前記パラメータにしたがって、プリンタに送るプリント・データを生成する動作モードを制御することを特徴とするプリント・システム。

【請求項2】 前記プリント処理は、前記描画データをスプール・ファイルにスプールすることを含み、

前記パラメータは、前記描画データを前記スプール・ファイルに格納する前に生成し、

前記生成したパラメータは、前記スプール・ファイルに前記描画データとともに保持しておき、

前記スプール・ファイルからデスプールした後に、前記パラメータをスプール・ファイルから抽出することを特徴とする請求項1記載のプリント・システム。

【請求項3】 前記プリント処理は、前記描画データをスプール・ファイルにスプールすることを含み、

前記パラメータは、前記描画データを前記スプール・ファイルに格納する前に生成し、

前記生成したパラメータは、前記スプール・ファイルとは別に保持しておき、

前記スプール・ファイルから描画データをデスプールした後に、対応するページの前記パラメータを抽出することを特徴とする請求項1記載のプリント・システム。

【請求項4】 前記プリント処理は、前記描画データをスプール・ファイルにスプールすることを含み、

前記パラメータは、描画データをスプール・ファイルに格納した後に生成し、

前記生成したパラメータは、前記スプール・ファイルとは別に保持しておき、

前記スプール・ファイルからデスプールした後に、前記パラメータを抽出することを特徴とする請求項1記載のプリント・システム。

【請求項5】 前記プリント処理は、プリント・データを生成するプリンタ・ドライバを含み、前記パラメータにしたがって制御される描画モードは、前記プリンタ・ドライバの動作モードであることを特徴とする請求項1～4いずれか記載のプリント・システム。

【請求項6】 前記プリンタ・ドライバは、前記アプリ

ケーションが作成した描画データから、プリンタが印刷するドット・パターンを作成することを特徴とする請求項1～5いずれか記載のプリント・システム。

【請求項7】 前記プリント処理の前記描画データの各ページ内容を検出す場合、各ページのデータの属性が、テキスト・データの属性が主であるのか、イメージが主であるのかを判定することを特徴とする請求項1～6いずれか記載のプリント・システム。

【請求項8】 前記プリンタ・ドライバの前記パラメータにより制御される動作モードには、前記ドット・パターンの処理解像度制御を含むことを特徴とする請求項1～7いずれか記載のプリント・システム。

【請求項9】 前記動作モードには、ベクトル・データのラスタライズ処理のビット長制御を含むことを特徴とする請求項8記載のプリント・システム。

【請求項10】 前記ラスタライズ処理のビット長制御には、他のイメージ・データのビット長と整合させるビット長整合を含むことを特徴とする請求項9記載のプリント・システム。

20 【請求項11】 ホスト・コンピュータ上の、アプリケーションの生成した描画データからプリンタへのプリント・データを形成し、プリンタに転送して印刷するプリント・システムにおけるプリント処理方法において、前記描画データの各ページ内容を検出し、検出した結果に応じて、各ページごとの描画モードを判定し、各ページごとに描画モードを指定するパラメータを生成し、生成した前記パラメータを保持し、

30 各ページ処理に同期して、保持していた前記パラメータにしたがって、プリンタに送るプリント・データを生成する動作モードを制御することを特徴とするプリント処理方法。

【請求項12】 ホスト・コンピュータ上の、アプリケーションの生成した描画データから、プリンタへのプリント・データを形成し、プリンタに転送して印刷するプリント・システムにおけるプリント処理を行うプログラムを格納した記録媒体において、前記プリント処理は前記描画データの各ページ内容を検出し、

40 検出した結果に応じて、各ページごとの描画モードを判定し、各ページごとに描画モードを指定するパラメータを生成し、生成した前記パラメータを保持し、

各ページ処理に同期して、保持していた前記パラメータにしたがって、プリンタに送るプリント・データを生成する動作モードを制御することを特徴とするプリント処理を行うプログラムを格納した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ホスト・コンピュータとプリンタとを接続し、ホスト・コンピュータ上のアプリケーションからの印刷データをプリンタに送って印刷するプリンタ・システムに関し、特に、ページごとの描画しようとするデータ属性に応じてプリンタ処理を自動的に変えることができるプリント・システムに関する。

【0002】

【背景の技術】ホスト・コンピュータからプリンタに対して、プリントするためのデータを受け渡す場合、スプーリング (pooling) により行うことは、従来から行われている。

【0003】このスプーリングの処理の例を図13に示す。図13において、101はホスト・コンピュータ (PC: パーソナル・コンピュータ) 内の処理であることを示しており、102はプリンタへ出力されることを示している。アプリケーション103は、生成したプリント・ジョブ・データを描画制御システム104に順次渡す。

【0004】描画制御システム104は、プリント・ジョブ・データをプリンタ・ドライバ111の描画処理できる形式に変換してスプーラ105に渡す。このデータは論理的な描画オブジェクト、及びプリンタに対する描画制御情報から成る。描画制御システム104は通常システム・レベルのプログラムで、例えばWindowsではGDIがこれに相当する。

【0005】スプーラ105は、描画制御システム104の出力をページ・パラメータ判定部107を介してスプール・ファイル108にセットするとともに、キュー制御106にスプール完了を通知してプリント・ジョブのキューイング処理を行わせる。キュー制御106はプリント処理の実行順をプリント・ジョブ単位で管理する。キュー制御106において、スプーラ105からプリント・ジョブをスプール・ファイル108へセットした旨の通知を受け取ると、そのプリント・ジョブをプリント・キューに登録する。また、デスプーラ109を起動して、通常キューの先頭 (最も古いプリント・ジョブ) からプリント・ジョブの出力を実行させる。

【0006】デスプーラ109はキュー制御106に指示されたプリント・ジョブのスプール・ファイルを取りだし、プリンタ・ドライバ111へ渡す。プリンタ・ドライバ111は、描画制御システム104の生成した論理的な描画オブジェクト、及びプリンタに対する描画制御情報をプリンタが認識可能な物理的な描画情報、制御情報に変換処理し、I/O制御部112を介してプリンタ102へ送信する。

【0007】I/O制御部112は、プリンタ・ドライバ111の生成したデータ・ストリームをプリンタへ送信するためのデータ・バッファリング、I/O制御を行う。

【0008】ユーザインターフェース1101は、プリンタ・ドライバの動作モードをユーザが指示するためのインターフェースで、通常、アプリケーションがプリント処理時にダイアログボックスを開き、この中でユーザが動作モードやその他のドライバ処理に関するパラメータを指示する。

【0009】ここで、プリンタ・ドライバ111の処理解像度とプリンタ102の実際に印刷する解像度についていえば、従来はプリンタ・エンジンの解像度でドライバ処理を行うことが一般的であった。しかし、近年のプリンタ・エンジンの高解像度化にともない、実際に印刷を行う解像度でドライバ処理を行うと処理時間が長くなり、またイメージ・データなど階調性を必要とするデータでは、単なる高解像度処理では高品位な印刷表現が困難となってきた。このため、イメージ・データに対しては、ドライバのラスター処理では低解像度処理を行い、プリンタ・エンジンの高解像度化は濃度パターン法など階調表現を有効に利用することが考えられる。

【0010】たとえば、プリンタ・エンジンが600 dpiであるならば、プリンタ・ドライバは、エッジ品位の必要とされるテキスト、グラフィックスのプリント・ジョブ印刷に対しては、600 dpiの解像度でラスター処理して印刷を行う。また、階調表現の必要なイメージ・データのプリント・ジョブに対しては、プリンタ・ドライバは300 dpiでラスター処理を行い、プリンタ・エンジンの2×2ドット・ブロック毎に階調表現を行うことが可能となる。

【0011】ユーザ・インターフェース1101を用いて、このようなラスター処理解像度をプリント・ジョブ毎に切り換えることができる。しかし、プリント・ジョブ毎に処理解像度、ラスタライズ・ビット長は固定であり、ジョブ内で切り換えることはできなかった。

【0012】

【発明が解決しようとする課題】上記の従来例では、各種のアプリケーションの生成したデータは、同一プリント・ジョブ内において複数の描画オブジェクトから構成されている。各描画オブジェクトは下記のいずれかの属性を持つ。

【0013】i) イメージ

ii) グラフィックス

iii) テキスト

ジョブ内は同一解像度で描画オブジェクトをレンダリングし、その表現手段 (色変換処理、レンダリング解像度、ドットパターン形成) は、いずれのオブジェクト属性においてもバランスがとれるよう設計されている。

【0014】しかしながら、従来例では以下の問題があった。

【0015】(1) テキストの品位を保つため、プリント・ジョブ全体を高解像度でラスタライズすると、ラスタライズ処理自体、色変換処理に時間がかかるとともに

に、メモリーの使用量が増加する。

【0016】(2) 处理速度、階調性を優先して低解像度でラスター処理を行うと、テキストなどでエッジ品位が低下し、ジャギーが目立つ。

【0017】(3) 描画オブジェクトの中には、解像度重視のもの(テキスト、グラフィクス)と階調(色表現)重視のものの2種があり、それらが混在するプリント・ジョブではその表現品位と処理速度との両立は困難である。

【0018】

【課題を解決するための手段】上記問題点を解決するため、請求項1記載の発明は、ホスト・コンピュータ上のアプリケーションの生成した描画データから、プリンタへのプリント・データを形成し、プリンタに転送して印刷するプリント処理を含むプリント・システムにおいて、前記プリント処理は、前記描画データの各ページ内容を検出し、検出した結果に応じて、各ページごとの描画モードを判定し、各ページごとに描画モードを指定するパラメータを生成し、生成した前記パラメータを保持し、各ページ処理に同期して、保持していた前記パラメータにしたがって、プリンタに送るプリント・データを生成する動作モードを制御する。

【0019】これにより、同一プリント・ジョブ内部においても、各ページ毎に描画内容に応じて印刷結果の表現品位と処理速度とが最適化することができる。

【0020】請求項2記載の発明では、前記プリント処理は、前記描画データをスプール・ファイルにスプールすることを含み、前記パラメータは、前記描画データを前記スプール・ファイルに格納する前に生成し、前記生成したパラメータは、前記スプール・ファイルに前記描画データとともに保持しておき、前記スプール・ファイルからデスプールした後に、前記パラメータをスプール・ファイルから抽出する。

【0021】これにより、スプール・データにパラメータ(描画制御情報)を埋め込んで管理し、スプール・データを取り出す時にその中からパラメータを取りだし、それに応じてプリント処理を切り換えることができる。

【0022】請求項3記載の発明では、前記プリント処理は、前記描画データをスプール・ファイルにスプールすることを含み、前記パラメータは、前記描画データを前記スプール・ファイルに格納する前に生成し、前記生成したパラメータは、前記スプール・ファイルとは別に保持しておき、前記スプール・ファイルから描画データをデスプールした後に、対応するページの前記パラメータを抽出する。

【0023】これにより、スプール・データとは別個にパラメータ(描画制御情報)を管理し、スプール・データを取りだすときに、それと同期してパラメータを取りだし、それに応じてプリント処理を切り換えることができる。

【0024】請求項4記載の発明では、前記プリント処理は、前記描画データをスプール・ファイルにスプールすることを含み、前記パラメータは、描画データをスプール・ファイルに格納した後に生成し、前記生成したパラメータは、前記スプール・ファイルとは別に保持しておき、前記スプール・ファイルからデスプールした後に、前記パラメータを抽出する。

【0025】これにより、スプール・データのスプール完了後、スプール処理とは別個にスプール・データ内容をチェックし、描画制御を判定してパラメータ(描画制御情報)を生成し、スプール情報とは別個にパラメータを管理している。スプール・データを取りだし、描画処理を行うときに、それと同期してパラメータを取りだし、それに応じてプリント処理を行うことができる。

【0026】請求項5記載の発明では、前記プリント処理は、プリント・データを生成するプリンタ・ドライバを含み、前記パラメータにしたがって制御される描画モードは、前記プリンタ・ドライバの動作モードであることを特徴とする請求項1～4いずれか記載のプリント・システムである。

【0027】これにより、動作モード切り換え機能付プリンタ・ドライバを、パラメータにより、このプリンタ・ドライバ処理の動作モードを自動的に制御することができる。

【0028】請求項6記載の発明では、前記プリンタ・ドライバは、前記アプリケーションが作成した描画データから、プリンタが印刷するドット・パターンを作成する。

【0029】これにより、ホスト・コンピュータに接続されたプリンタ側の処理が簡単になる。

【0030】請求項7記載の発明では、前記プリント処理の前記描画データの各ページ内容を検出する場合、各ページのデータの属性が、テキスト・データの属性が主であるのか、イメージが主であるのかを判定する。

【0031】これにより、各ページのデータの属性に適した動作モードで自動的に印刷することができる。

【0032】請求項8記載の発明では、前記プリンタ・ドライバの前記パラメータにより制御される動作モードには、前記ドット・パターンの処理解像度制御を含む。

【0033】これにより、印刷されるデータの解像度を各属性(テキストやイメージ)に適切になるように自動的に制御して印刷することができる。

【0034】請求項9記載の発明では、請求項8記載の発明において、前記動作モードには、ベクトル・データのラスタライズ処理のピット長制御を含む。

【0035】これにより、解像度制御と連動してベクトル・データのラスタライズのピット長を切り替えることができるため、高解像度処理ページにおいても、ラスタライズ長を制御することにより、処理時間を短縮することができる。

【0036】請求項10記載の発明では、前記ラスタライズ処理のビット長制御には、他のイメージ・データのビット長と整合させるビット長整合を含む。

【0037】これにより、ペクトル・データに対しても、イメージ・データとの同一のラスター・オペレーションが可能になる。

【0038】請求項11記載の発明では、ホスト・コンピュータ上のアプリケーションの生成した描画データから、プリンタへのプリント・データを形成し、プリンタに転送して印刷するプリント・システムにおけるプリント処理方法において、前記描画データの各ページ内容を検出し、検出した結果に応じて、各ページごとの描画モードを判定し、各ページごとに描画モードを指定するパラメータを生成し、生成した前記パラメータを保持し、各ページ処理に同期して、保持していた前記パラメータにしたがって、プリンタに送るプリント・データを生成する動作モードを制御する。

【0039】これにより、同一プリント・ジョブ内部においても、各ページ毎に描画内容に応じて印刷結果の表現品位と処理速度とが最適化することができる。

【0040】請求項12記載の発明は、ホスト・コンピュータ上のアプリケーションの生成した描画データから、プリンタへのプリント・データを形成し、プリンタに転送して印刷するプリント・システムにおけるプリント処理を行うプログラムを格納した記録媒体において、前記プリント処理は、前記描画データの各ページ内容を検出し、検出した結果に応じて、各ページごとの描画モードを判定し、各ページごとに描画モードを指定するパラメータを生成し、生成した前記パラメータを保持し、各ページ処理に同期して、保持していた前記パラメータにしたがって、プリンタに送るプリント・データを生成する動作モードを制御する。

【0041】この記録媒体で、各ホスト・コンピュータに対して上記機能をインストールすることができる。

【0042】

【発明の実施の形態】図面を用いて、本発明の実施の形態を説明する。

【0043】まず、本発明の実施の形態が実行されるシステム例の概略について説明する。

【0044】図1は、本発明の処理が行われているシステムを説明する図である。この図1において、1はホスト・コンピュータ(PC:パーソナル・コンピュータ)で、アプリケーションやプリンタ・ドライバが実行される。2はプリンタで、ホスト・コンピュータ1に接続され、送られてきたプリント・データを実際に印刷している。3はフロッピーディスクで、例えば、プリンタ・ドライバ等のプログラムを格納しており、これらのプログラムをホスト・コンピュータにインストールできるものである。

【0045】図2は、本発明の処理が行われているホス

ト・コンピュータ1の構成の例を示すブロック図である。図2において、10はCPUで、ホスト・コンピュータにおける処理を実行している。12はROMで、BIOS等のプログラムが予め格納されているものである。14は主記憶で、プログラム、データが格納されCPU10により使用される。16はディスプレイ・コントローラで、ディスプレイ18とのインターフェースを行い、ディスプレイ18に表示を行っている。20は、SCSI等のディスク・コントローラで、ハード・ディスク・ドライブ22等とのインターフェースを行っている。ハード・ディスク22には、プログラム、データ等が格納されており、必要に応じて主記憶に呼び出されてCPUに使用されている。24はフロッピィ・ディスク・コントローラで、フロッピィ・ディスク・ドライブ26とのインターフェースを行っている。28はプリンタ・コントローラでプリンタ30とのインターフェースを行っている。32はキーボード・コントローラでキーボード34とのインターフェースを行っている。36はマウス・コントローラでマウス38とのインターフェースを行っている。ディスプレイ18、キーボード34およびマウス38で、ユーザ・インターフェースを構成している。

【0046】本発明は、上記のシステム上で実施されている。

【0047】【第1の実施の形態】本発明の第1の実施の形態を図3ないし図7に示す。

【0048】図3はプリント処理の流れを表わしている。1はホスト・コンピュータ(PC)、2はプリンタである。これは、図1、図2で示したホスト・コンピュータおよびプリンタと同様のものである。図3は、ホスト・コンピュータで実行されているアプリケーション103で生成した印刷データをプリンタ2に送り、印刷を実行する処理を表している。図13で示した従来の処理と同様の処理は、同じ符号を用いている。

【0049】さて、アプリケーション103は、生成したプリント・ジョブ・データを描画制御システム104に順次渡す。描画制御システム104は、プリント・ジョブ・データをプリンタ・ドライバ111の描画処理でできる形式に変換してスプーラ105に渡す。このデータは、論理的な描画オブジェクト、及びプリンタに対する描画制御情報から構成されている。描画制御システム104は通常システム・レベルのプログラムであり、例えばWindowsではGDIがこれに相当する。

【0050】スプーラ105は、描画制御システム104の出力をページ・パラメータ判定部107を介してスプール・ファイル108にセットするとともに、キューリング部106にスプール完了を通知してプリント・ジョブのキューリング処理を行わせる。このスプール・ファイルは、図2のハード・ディスク上に作成され、格納され

【0051】ページ・パラメータ判定部107は、スプーラ105の出力を順次チェックし、ページ毎の描画指示を行うパラメータ（以下ページ・パラメータと称する）をスプール・ファイルの中に埋め込む。

【0052】キュー制御106は、プリント処理の実行順をプリント・ジョブ単位で管理する。キュー制御106において、スプーラ105からプリント・ジョブをスプール・ファイル108へセットした旨の通知を受け取り、そのプリント・ジョブをプリント・キューに登録する。また、デスプーラ109を起動して通常キューの先頭（最も古いプリント・ジョブ）からプリント・ジョブの出力を実行させる。

【0053】デスプーラ109はキュー制御106に指示されたプリント・ジョブのスプール・ファイルを取りだし、ページ・パラメータ制御部110へ渡す。

【0054】ページ・パラメータ制御部110は、プリント・ジョブの内容をチェックし、そのジョブの各ページ毎に上述したページ・パラメータ判定部107が判定した描画指示に従って描画の指示を行うとともに、その指示情報（ページ・パラメータ）を削除し、プリンタ・ドライバ111へジョブのデータを順次渡す。これにより、プリンタ・ドライバ111の受け取るデータはスプーラ105が出力したものと同一となる。

【0055】プリンタ・ドライバ111は、描画制御システム104の生成した論理的な描画オブジェクト、及びプリンタに対する描画制御情報を、プリンタが認識可能な物理的な描画情報、制御情報を変換処理し、I/O制御部112を介してプリンタ102へ送信する。

【0056】I/O制御部112は、プリンタ・ドライバ111の生成したデータ・ストリームをプリンタへ送信するためのデータバッファリング、I/O制御を行う。

【0057】スプリーリング、キュー制御、デスプリーリング、ページ・パラメータ制御、及びプリンタ・ドライバのさらに詳細な処理を図4～図7に示す。

【0058】まず、スプーラ105の処理の例を図5に示す。スプーラ105は、描画制御システム104からの描画情報をスプール・ファイルにセットする（801）。次に、キュー制御が起動しているか確認する（802）。起動していないければ、キュー制御を起動し（803）、キュー制御へセット完了を通知して、そのプリント・ジョブをキューに登録する（804）。

【0059】図4には、ページ・パラメータ判定107の具体的な例として、各ページ毎の処理解像度をページ・パラメータとする例を示している。図4において、解像度判定201がこの処理を行っている。この解像度判定201は、スプーラから渡される描画情報を順次チェックし、ページ毎に解像度を判定して、その指示情報をスプール・ファイル108の中に埋め込む処理を行っている。

【0060】この解像度処理の詳細な例を図7に示す。

図7において、各描画オブジェクトの属性をチェックしてゆき（901）、ページの終わりに達した時点（902）でそのページの処理解像度を判定する（903）。その判定した情報をスプール・ファイルのページの区切りの先頭に埋め込む（904）。その処理をプリント・ジョブの終了まで繰り返す（905）ことにより、各ページの処理解像度を判定していく。

【0061】ここで行っている解像度判定は各ページの内容に依存する。例えば、テキスト主体のページであればエッジのキレを良くするため高解像度処理を行い、イメージ中心であれば通常解像度での処理を行うと判定する。

【0062】ページ検出／解像度制御部202は、図3のページ・パラメータ制御部110の具体例である。ここでは、デスプーラ109から渡されたスプール・ファイル情報のページ区切りを検出し、そのページ毎の処理解像度指示情報を取り出してプリンタ・ドライバ内の各処理モードを切り換える処理を行っている。

【0063】このページ検出／解像度制御部202の詳細な処理については、図6に示すフローチャートを用いて説明する。図6において、まず、スプール情報を取得（1001）し、ページ・スタートを検出したら（1002）、次のスプール情報取得（1003）で処理解像度情報であるかをチェックする（1004）。これは、図4（又は図7）で説明した解像度判定201（又は905）においてセットされた、ページの区切りの先頭の解像度情報（処理解像度情報）を検出することに対応している。処理解像度情報が検出されると、プリンタ・ドライバ各部をその処理モードにセットし（1005）、処理解像度情報を削除する（1007）。そうでなければ、プリンタ・ドライバ各部をあらかじめ決められたデフォルト・モードにセットする（1006）。

【0064】ページ・スタート直後でなく、処理解像度情報を現れた場合（1008におけるY）は、その情報を無効であるとして削除する（1007）。取得したスプール・ファイル情報が処理解像度情報でなければ（1008におけるN）、スプール・ファイル解析203へ渡す（1009）。

【0065】上記の処理をプリント・ジョブ・エンドまで繰り返す（1010）。これがページ検出／解像度制御202の処理である。

【0066】さて、図4において、スプール・ファイル解析203以下は、図3におけるプリンタ・ドライバ111の内部での処理である。

【0067】スプール・ファイル解析203は、スプール・ファイル内容をチェック、分類し、そのカテゴリーに応じて各処理モジュールに振り分ける。プリンタ制御情報であればプリンタ制御204へ、描画処理情報であればオブジェクト判定205へ振り分ける。

【0068】プリンタ制御204は、プリンタ制御情報の論理的な内容に応じて具体的なプリンタの制御情報を変換し、プリンタコマンド制御211を介してプリンタ制御コマンドをプリンタ102へ送信して制御する。このとき、プリンタ・コマンド制御211は実際にプリンタ102が受け取れるコマンドを生成する。

【0069】オブジェクト判定205は、描画オブジェクトがベクトル・データであればラスタライザ206へ、ラスター・データであれば解像度変換207へ渡す。ラスタライザ206および解像度変換207は、各描画オブジェクト毎にプリンタ102へ送信する解像度のラスター・データを生成する。ラスター・オペレーション208は、ラスター・バンド・バッファ上に実際の描画イメージを形成する。

【0070】バンド毎の描画が完了すると、その描画イメージは、色処理/コード化処理210を経て、プリンタコマンド制御211でプリンタへ送信するコマンド/データに変換され、プリンタ102に送信される。

【0071】色処理/コード化処理210は、論理的な色空間情報であるバンド・バッファ209上の描画情報を、プリンタ・エンジンに合わせた物理色空間データに変換するとともに、実際にプリンタに送るコードに変換する。これらの処理には、変換、2値化処理、圧縮処理なども含まれる。

【0072】プリンタコマンド制御211は、描画ラスター・データ、プリンタ制御情報をプリンタ102が受け取れるコマンドに変換して、図3におけるI/O制御112に渡す。

【0073】なお、図6において、処理の1005および1006がプリンタ・ドライバ各部に解像度の応じた処理モードにセットしているが、これは、図4において、ページ検出/解像度制御202からの線が206～211の処理に対して入力している様子で表している。

【0074】上記で説明したように、ページ検出/解像度制御202は、各ページの処理解像度に応じて、ラスタライザ206、解像度変換207、ラスター・オペレーション208の処理解像度を切り換えている。また、必要に応じて、色処理/コード化処理210の動作モードを切り換えている。さらに、プリンタ・コマンド制御211を介してコマンドを発行し、そのページの処理解像度、転送データ解像度をプリンタ102へ通知している。

【0075】プリンタ102は、このコマンドに応じて、処理データの解像度を切り換えて処理を行う。

【0076】以上述べた本実施形態では、ページ・パラメータをスプール・ファイルに埋め込むことにより、ページ・パラメータ管理及び各ページ処理に同期したページ毎の描画制御を容易に行うことが可能となる。

【0077】【第2の実施形態】本発明の第2の実施形態を図8および図9に示す。第1の実施形態の図3およ

び図4と同様の処理は、同一の番号で示している。

【0078】図8を用いて、第2の実施形態の概略を説明する。図8において、ページ・パラメータ判定301で判定したページ毎の処理の切り替え情報であるページ・パラメータ302は、スプール・ファイルと分離して保持している。スプール・ファイルをプリンタ・ドライバ111が処理するとき、ページ・パラメータ303は、スプール・ファイルとは別に保持されているページ・パラメータ302にアクセスして制御を行う。

【0079】図8の処理の詳細な例を図9に示す。図3および図4と同様の処理には同一の番号で示してある。図9に示す処理では、第1の実施形態の図4と同様に、ページ毎の処理を切り換えるパラメータとして処理解像度を例にしている。

【0080】解像度判定401は、第1の実施例の201と同様の処理を行うが、解像度情報はスプール・ファイルの中に埋め込むのではなく、別の保持手段で保持する。これは別ファイルでも良いし、メモリー上にアロケートされた構造体でもよい。

【0081】スプール・ファイル解析404は、デスプーラから渡されるスプール・ファイル内容を解析し、第1の実施例の場合と同様に内容毎に各処理ブロックに振り分ける。ここでページの区切りを検出するとその情報をページ検出/管理405へ渡し、このページ検出/管理405は解像度制御403を起動する。

【0082】解像度制御403は、そのプリント・ジョブの解像度情報402へアクセスし、その内容に従って各処理ブロック206～211のモードを切り換える。

【0083】以上説明したように、本実施形態では、ページ・パラメータをスプール・ファイルと別個に管理している。このため、ページ・パラメータへのアクセス、操作が容易になるので、ページ・パラメータ判定と異なる処理を行わせる場合には、このような処理を簡単に行うことが可能となる。

【0084】【第3の実施形態】本発明の第3の実施形態を図10および図11に示す。第1実施形態および第2実施形態の図である図3、図4、図8および図9と同様の処理は、同一の番号で示している。

【0085】この実施形態3では、ページ・パラメータの判定/制御は、スパーリング処理とは独立したタスク(プロセス)として処理を行う。

【0086】スプーラ105は描画制御システム104の出力をスプール・ファイル108にセットするとともにキュー制御501にスプール完了を通知してプリント・ジョブのキューイング処理を行わせる。キュー制御501はページ・パラメータ判定/制御502を起動し、ページ毎のパラメータを抽出させてからデスプーラ109を起動し、プリンタ・ドライバへ情報を渡す。

【0087】図10の処理の詳細を図11に示す。この例では、第1および第2の実施形態の詳細と同様に、ペ

ージ毎の処理を切り換えるパラメータとして処理解像度を例にしている。

【0088】図11において、キュー制御501は、スプール105がスプール・ファイルのセット完了後に起動され、解像度判定601を起動する。解像度判定601は、スプール・ファイルの内容のスキャンおよびチェックを行い、各ページ毎の解像度情報を判定して、解像度情報402としてページ・パラメータ判定/制御502の内部で保持する。プリント・ジョブ全体のスキャン、チェックを終了すると次に、キュー制御501は、デスプーラ109を起動し、スプール・ファイル内容をプリンタ・ドライバ111に渡して、プリント処理を行わせる。

【0089】第2の実施形態と同様に、プリンタ・ドライバ111のスプール・ファイル解析404は、ページの区切り情報をページ検出/管理405へ渡す。ページ検出/管理405は、ページ・パラメータ判定/制御502の解像度制御403を起動する。

【0090】解像度制御403は、そのプリント・ジョブの解像度情報402へアクセスし、その内容に従って各処理ブロック206~211のモードを切り換える。

【0091】以上説明した本実施形態では、ページ・パラメータ判定/制御をスਪーリング処理と別タスクで行っている。このため、ページ毎の処理切り換えを行わない従来の処理系として独立してページ・パラメータ判定/制御機能を付加しやすい。また、プリント処理のアプリケーション開放時間もページ・パラメータ判定分早くできる。

【0092】【第4の実施形態】本発明の第4の実施形態を図12に示す。図12において、第1~第3の実施形態の図2、図3、図8~図11と同一の構成要素は同一の番号で示している。

【0093】本実施形態は、第1の実施形態に類似しているが、本実施形態においては、ページ・パラメータとして解像度とラスタライズのビット長とを制御している。

【0094】図12において、ページ検出701は、第1の実施形態と同様にページ区切りを検出し、解像度判定201がスプール・ファイル108に埋め込んだページ毎の処理解像度情報を解像度制御702とラスタライズ・ビット制御703に渡している。解像度制御702では、ドライバ各部の処理ブロック206~211の動作モードを切り換え、解像度を制御している。ラスタライズ・ビット制御703は、解像度制御702と連動する形で、ベクトル・データをラスタライズする際の画素毎のビット長（ビット深さ）を制御する。

【0095】ラスタライズ・ビット制御703は、高解像度処理を行う場合は、そのページ内容は、テキストもしくは色数の少ないグラフィックス主体のページであると仮定したものである。このラスタライズ・ビット長を高

解像度処理時に短くすることにより、ラスタライズ実行時間を短縮することができる。

【0096】オブジェクト判定205は、描画オブジェクトをビットマップデータとベクトル・データに振り分け、前者は解像度変換207へ、後者はラスタライザ206へ渡す。ラスタライザ206はラスタライズ・ビット制御703の指示ビット長でラスタライズを行う。ビット整合704を用いて、ラスタライザ206出力のビット長を解像度変換207出力のラスター・データと同一のビット長に変換することもできる。このように、ラスター・データが同一であると、同一のラスター・オペレーションが両者のデータに対して同じように行うことができる。

【0097】これ以降の動作は、第1の実施形態と同様である。

【0098】なお、このラスタライズ・ビット制御について、第1の実施形態と同様の処理について説明したが、この分野の通常の知識を有する者であれば、上記第2および第3の実施形態においても適用でき、実施できることは、自明である。

【0099】以上述べた本実施形態では、ページ・パラメータの解像度制御と連動して、ベクトル・データのラスタライズのビット長を切り換えることができる。このため、高解像度処理ページにおいても、ラスタライズ・ビット長を短くすることにより、処理時間の増加を最少に留めることができる。またページ毎の描画内容に応じて、エッジの品位、階調性、処理時間をきめ細かく制御できる。

【0100】本発明は、システムあるいは装置にプログラムを供給することによって達成される場合にも適用される。

【0101】この場合、図1に示したように、本発明を達成するためのプログラムを格納した記憶媒体（例えば、フロッピィ・ディスク3やハード・ディスク22）をシステムあるいは装置が読み出すことによって、上記説明した本発明の機能を実現し、本発明の効果を享受することができる。この記録媒体には、フロッピィ・ディスクやハード・ディスク以外にも、例えばCD-R/O、磁気テープ、光磁気ディスク、メモリ・カード、ROMカセット等も含まれる。

【0102】

【発明の効果】以上述べたように、本発明においては、異なる属性の複数の描画オブジェクトからなる複数ページのプリント・ジョブを印刷する場合において、ページ毎の描画内容に応じてプリンタ・ドライバの処理を切り換えることが可能となり印刷結果の表現品位と処理速度との最適化が可能となる。

【0103】ページ毎の描画内容に応じたプリンタ・ドライバ制御情報をスプール・データに埋め込んで管理することにより、プリンタ・ドライバ制御情報管理及び各

15

ページ処理に同期したページ毎の描画制御を容易に行うことが可能となる。

【0104】ページ毎の描画内容に応じたプリンタ・ドライバ制御情報をスプール・データと別個に保持することにより、プリンタ・ドライバ制御情報へのアクセス、操作が容易になるため、プリンタ・ドライバ制御判定と異なる処理を行わせたい場合にそのための処理を簡単に行うことが可能となる。

【0105】ページ毎の描画内容に応じたプリンタ・ドライバ制御判定／制御をスプーリング処理と別タスクで行うため、ページ毎の処理切り換えを行わない従来の処理系と独立してプリンタ・ドライバ制御判定／制御機能を付加しやすい。また、プリント処理のアプリケーション開放時間がプリンタ・ドライバ制御判定分早くできる。プリンタ・ドライバ制御として、プリンタ・ドライバ処理像度を切り換えることにより、印刷出力のテキスト、グラフィックスのエッジ品位とイメージの階調性表現の両立が可能となる。

【0106】プリンタ・ドライバ制御としてベクトル・データのラスタライズ処理のビット長を切り換えることにより、高解像度処理の場合においても処理時間を短縮できる。

【図面の簡単な説明】

【図1】本発明が実施させるシステム構成を示す図である。

【図2】本発明が実施されるシステム構成を示すブロック図である。

【図3】本発明の第1の実施形態を説明する図である。

【図4】本発明の第1の実施形態を詳細に示す図である。

【図5】スプーラの処理を示す図である。

【図6】解像度判定の処理を示す図である。

【図7】解像度制御の処理を示す図である。

【図8】本発明の第2の実施形態を説明する図である。

【図9】本発明の第2の実施形態を詳細に示す図である。

【図10】本発明の第3の実施形態を説明する図である。

【図11】本発明の第3の実施形態を詳細に示す図である。

【図12】本発明の第4の実施形態を詳細に示す図である。

【図13】従来例を示す図である。

【符号の説明】

1 ホスト・コンピュータ

2 プリンタ

3 フロッピー・ディスク

10 CPU

12 ROM

16

1'4 主記憶 (RAM)

1'6 ディスプレイ・コントローラ

1'8 ディスプレイ

2'0 ディスク・コントローラ

2'2 ハード・ディスク

2'4 フロッピイ・ディスク

2'6 フロッピイ・ディスク

2'8 プリンタ・コントローラ

3'0 プリンタ

3'2 キーボード・コントローラ

3'4 キーボード

3'6 マウス・コントローラ

3'8 マウス

1'0'1 ホスト・コンピュータ

1'0'2 プリンタ

1'0'3 アプリケーションプログラム

1'0'4 描画制御システム

1'0'5 スプール

1'0'6 キュー制御

1'0'7 ページ・パラメータ判定

1'0'8 スプール・ファイル

1'0'9 デスプーラ

1'1'0 ページ・パラメータ制御

1'1'1 プリンタ・ドライバ

1'1'2 I/O制御

2'0'1 解像度判定

2'0'2 ページ検出／解像度制御

2'0'3 スプール・ファイル解析

2'0'4 プリンタ制御

3'0'5 オブジェクト判定

2'0'6 ラスタライザ

2'0'7 解像度変換

2'0'8 ラスター・オペレーション

2'0'9 バンドバッファ

2'1'0 色処理／コード化処理

2'1'1 プリンタコマンド制御

3'0'1 ページ・パラメータ判定

3'0'2 ページ・パラメータ

3'0'3 ページ・パラメータ制御

4'0'1 解像度判定

4'0'2 解像度情報

4'0'3 解像度制御

4'0'4 スプール・ファイル解析

4'0'5 ページ検出／管理

5'0'1 キュー制御

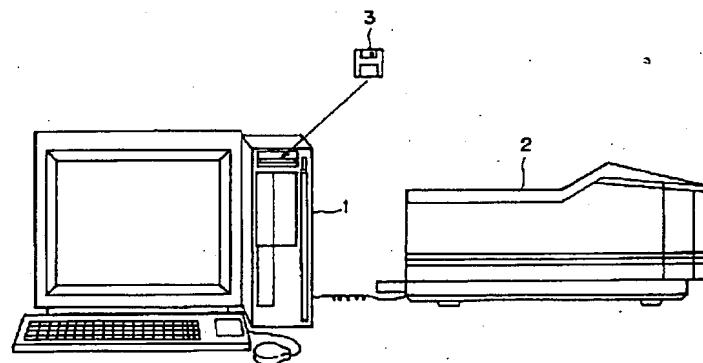
5'0'2 ページ・パラメータ判定／制御

7'0'1 ページ検出

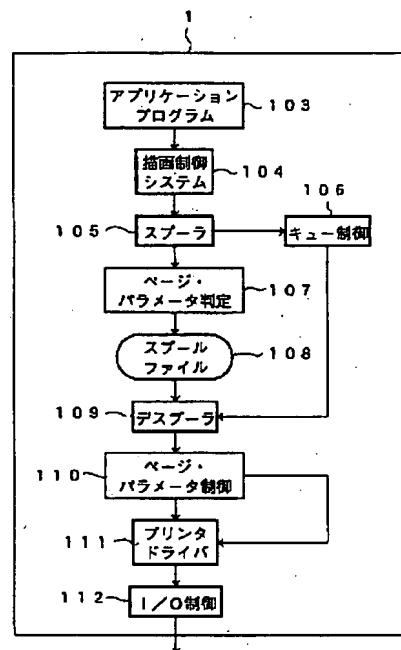
7'0'2 解像度制御

7'0'3 ラスタライズ・ビット制御

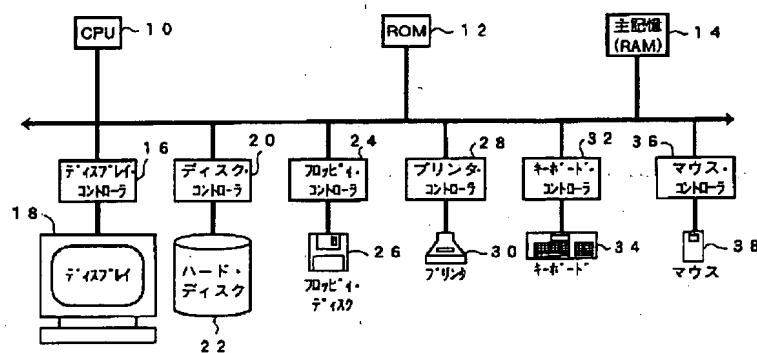
【図1】



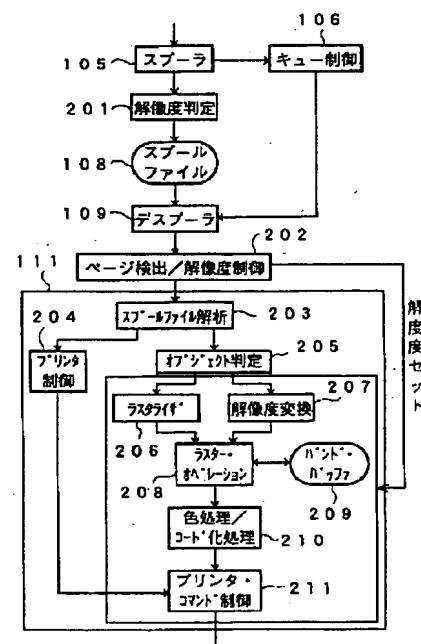
【図3】



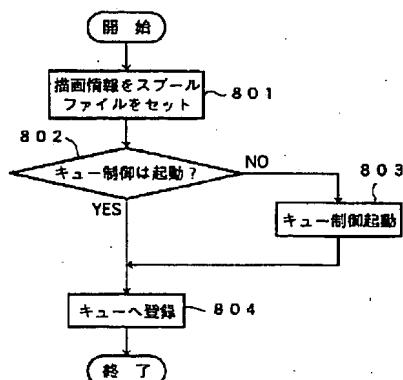
【図2】



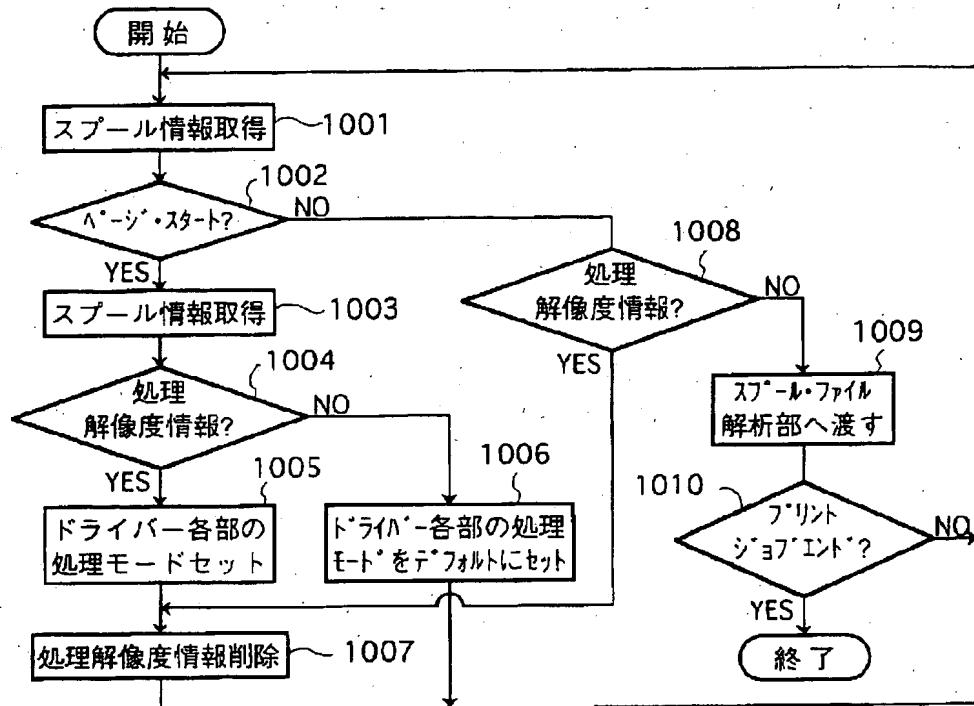
【図4】



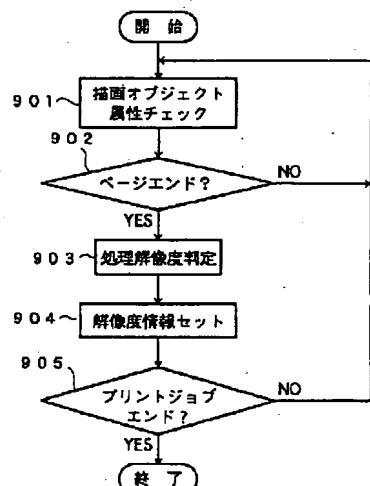
【図5】



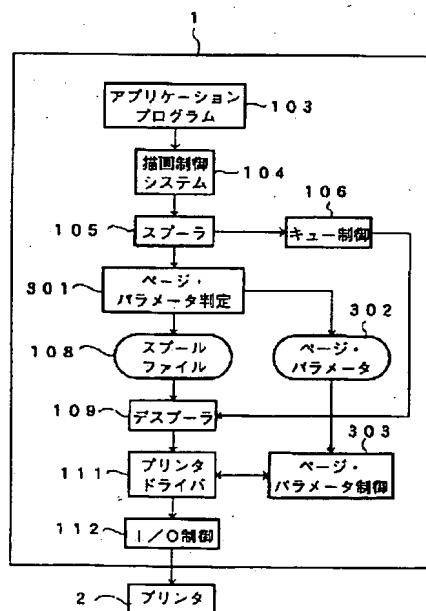
【図6】



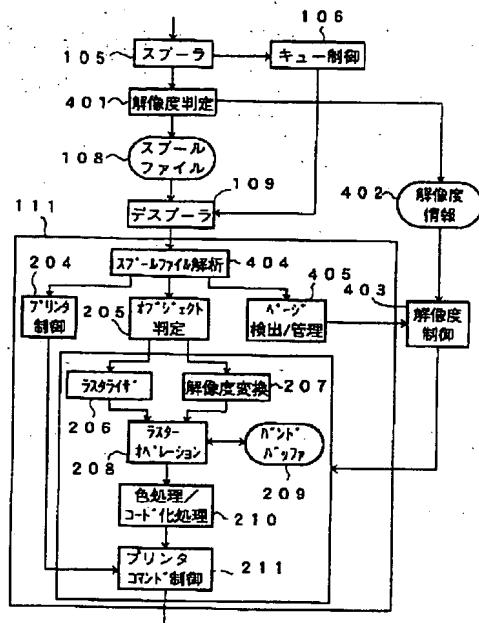
【図7】



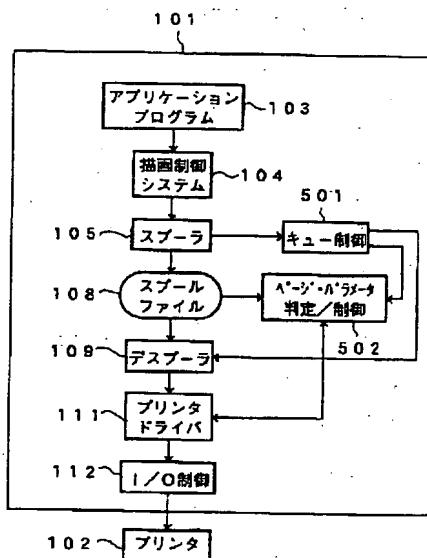
【図8】



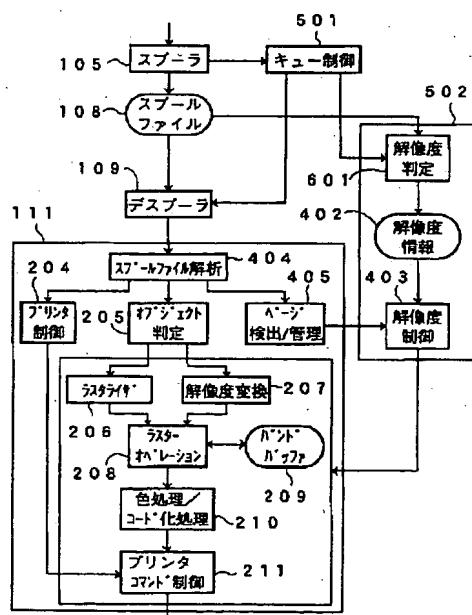
[図9]



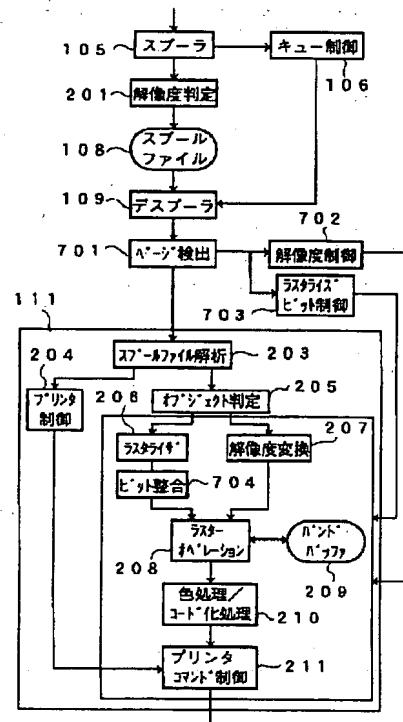
[図10]



【图 1-1】



【図12】



【図13】

